name	Array	Name of the array to create
size1	Number	Number of elements in the array or number of rows if size2 is specified
size2	Number	Number of columns in a two dimensional array

Creates name, an array of type Integer, with size1 number of elements. If specified, name will be a two-dimensional array with size1 columns and size2 rows. If name already exists, the array is resized and elements that are not cropped retain their value.

The size2 parameter is optional; if it is specified the commands create two-dimensional arrays. In this case, size1 specifies the number of rows and size2 specifies the number of columns in each array.

Each row in a two-dimensional array can be treated like an element.This means that you can insert and delete entire arrays in a two-dimensional array with the other array commands.

Creates Name, an array of type Pointer, with Size1 number of elements specified, Name will be a two-dimensional array with Size1 columns and Size2 rows. If Name already exists, the array is resized and elements that are not cropped retain their value.

An element 0 (array name0) is always created for an array, and is set to a null value of the array type. Use the Size f rray command to find the size of the array. the following line would delete all elements (except the 0 element), but leave the array defined:

```
ARRAY OINTER (◊Mine;0)
```

The following formula shows how to calculate the amount of memory used by a pointer array:

```
(1+number of elements)*16
```

Note: a few more bytes may be required to keep track of the selected element, the number of elements, and the array itself.

When a Pointer array is first created, its elements are empty values, that is NIL.

You refer to the elements by using curly braces. For example, MyArray2 refers to the second element in the array MyArray. For two-dimensional arrays you refer to individual elements by using two sets of curly braces. For example, MyArray35 refers to the third row and fifth column.

See also: Size f rray